(1) Japanese Patent Application Laid-Open No. JP63-318632(1988)

"High-Speed Square-Root Extraction Operation Unit "

The following is an extract relevant to the present application.

A high-speed square-root extraction operation device including an adder performing a binary addition of plural stages, a multiplexer of plural stages selecting an input signal of said adder, and a means for controlling said multiplexer based on a result of addition at the previous stage.

⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　昭63－318632

⑤Int.Cl.⁴　識別記号　庁内整理番号　⑬公開　昭和63年(1988)12月27日
G 06 F　7/552　B－7056－5B

審査請求　未請求　発明の数 1　（全5頁）

⑭発明の名称　高速開平演算装置

⑳特　願　昭62－155828
㉒出　願　昭62(1987)6月23日

⑦発明者　豊蔵　真木　大阪府門真市大字門真1006番地　松下電器産業株式会社内
⑦発明者　山田　晴保　大阪府門真市大字門真1006番地　松下電器産業株式会社内
⑦発明者　森　俊樹　大阪府門真市大字門真1006番地　松下電器産業株式会社内
⑦出願人　松下電器産業株式会社　大阪府門真市大字門真1006番地
⑦代理人　弁理士　中尾　敏男　外1名

明　細　書

1、発明の名称

高速開平演算装置

2、特許請求の範囲

複数段の2進数加算を行なう加算器を有する開平演算装置であって、前記加算器の入力信号を選択する複数段のマルチプレクサを具備し、前段の加算結果により前記マルチプレクサを制御する手段を有することを特徴とする高速開平演算装置。

3、発明の詳細な説明

産業上の利用分野

本発明はディジタル信号処理プロセッサにおいて、平方根の演算を行なう開平演算装置に関する。

従来の技術

第3図に一般的な非回復型の平方根の導出アルゴリズムを示す。（コンピュータの高速演算方式; Hwang, K. 細部ひさし訳 近代科学社 1980. 09参照）

以下、第3図に基づき、このアルゴリズムを説明する。

2つの正の2進数A, Qを次のように表す。

$$Q = \sqrt{A} = 0.q_1q_2 \cdots\cdots q_n$$

$$A = Q^2 = 0.a_1a_2 \cdots\cdots a_{2n-1}$$

まず、$D_0 = 0.01$とおきa_1a_2から減ずる。残りR_1が正の時$q_1 = 1$と置き、次の対a_3a_4をR_1に追加し、それから$D_1 = 0.q_101$を減ずる。R_1が負ならば$q_1 = 0$と置き、a_3a_4をR_1に追加し、$D_1 = 0.q_111$をそれに加える。この結果をR_2と置き、次の処理のための判断の対象となる。同様に処理が進められるが、一般にk番目のステップでは、k番目の平方根ビットq_kが決められたのち、次の操作を行なうことになる。

$R_k \leftarrow R_k \cdot a_{2k+1}a_{2k+2} - q_1q_2 \cdots\cdots q_k01$ $q_k = 1$のとき

$R_k \leftarrow R_k \cdot a_{2k+1}a_{2k+2} + q_1q_2 \cdots\cdots q_k11$ $q_k = 0$のとき

ここで追加操作"・"は1つ前の残余を2ビット式へシフトし、右端へ新しい対を補充することによって実現される。

このアルゴリズムを実現するためには、q_kが0か1かに応じて減算か加算かを選択することが可能な制御可能加算減算（Controlled Add －

Substract：ＣＡＳ）セルを用いる。ＣＡＳセルは、フルアダーとＥＸＯＲゲートから構成されている。８ビット２進数の平方根を４ビット２進数として計算するセル構成において、ＣＡＳセルを２０個必要とする。フルアダーを５個のゲートで構成すると、その他のインバータ３個と合わせて１２３個のゲートが必要となる。

発明が解決しようとする問題点

しかしながら上記のような構成では、前段の計算が終了するまで次段の計算は始まらないため、計算時間が遅く、また、ＣＡＳセルの構成によるため、多くの素子数が必要となっていた。

本発明はかかる点に鑑み、少ない素子数で、高速に平方根演算を行なう開平演算装置を提供することを目的とする。

問題点を解決するための手段

本発明は、複数段の２進数加算器とその入力信号を前段の加算結果により選択するマルチプレクサを備えた開平演算装置である。

作　用

本発明は前記した構成により、それぞれの段の加算を途中まで実行しておくことが可能となり、また、各段の演算は、加算（減算を２の補数の加算で実現している）のみでよく、演算を加算か減算かを制御する必要がないため、素子数の削減が可能となる。

実施例

以下本発明の一実施例を図面に基づいて説明する。まず、本発明の開平演算装置の基礎となる開平演算方式のアルゴリズムを第２図に示す。以下第２図に従って本演算方式のアルゴリズムを述べる。

1）初期設定として$k=1$，$R_0=a_1a_2$，$D_0=01$と置く。$R_0-D_0=a_1a_2-01$を計算し、R_1とする。（ステップ１，ステップ２）

2）残余R_kの値を調べる。（ステップ３）

ｉ）残余R_kが０または正のとき

$q_k=1$と置き、R_kに被開平数の次の２ビット分の$a_{2k+1}a_{2k+2}$を追加し、これをR_kに置く。（ステップ４）

ii）残余R_kが負のとき

$q_k=0$と置き、R_{k-1}に被開平数の次の２ビット分の$a_{2k+1}a_{2k+2}$を追加し、これをR_kに置く。（ステップ５）

3）$D_k=q_1q_2\cdots\cdots q_k01$と置き、$R_k-D_k$を計算し、$R_{k+1}$に置く。そして$k$に１を加える。（ステップ６，ステップ７）

4）繰返し回数kを調べる。（ステップ８）

ｉ）kがnを超えなければ2)へ戻り、実行を進める。

ii）kがnを超えれば、平方根Qを

$$Q=q_1q_2\cdots\cdots q_n$$

とし、処理を終える。（ステップ９）このとき残余は、R_{n-1}である。ここでnは必要とする平方根の桁数である。

以上で述べたアルゴリズムを実際のハードで構成するための前段階として実行手順を第４図の並算例に従って説明する。第４図では２進８桁の１００００００について開平演算を行ない、４桁の平方根と５桁の残余を求めている。まずa_1a_2に１１を加える。これは０１を減ずることの代わりに２の補数を利用している。この加算の結果キャリーがあれば、$q_1=1$、なければ$q_0=0$とする。今、キャリーがあり$q_1=1$とする。次に、加算の結果０１に$a_3a_4=00$を追加した０１００に$0q_101$の補数$1\bar{q}_111=1011$を加える。この和は１１１１でキャリーはない。このとき$q_2=0$とする。さらに次のステップでは、$q_2=0$であるため一段前の加算結果１００を対象とし、$a_5a_6=00$を追加した１００００に前と同様に１０１１１を加えた結果００１１１とキャリーを得、$q_3=1$とする。最後のステップでは０１１１に$a_7a_8=00$を追加した０１１１００に１０１０１１を加えた結果０００１１１とキャリーを得、残余００１１１及び平方根１０１１を得ることになり演算を終える。即ち$10000000_{(2)}=128_{(10)}$の平方根は$1011_{(2)}=11_{(10)}$であり残余は$111_{(2)}=7_{(10)}$となる。

次にこれまで述べてきた開平演算の実行手順を実現するための本発明の高速開平演算装置の一実

施例について以下で説明する。第１図が２進８桁の被開平数$a_1a_2a_3a_4a_5a_6a_7a_8$を２進４桁の平方根$q_1q_2q_3q_4$に計算する例を示すものである。第１図において、１０１～１１０はフルアダー、２０１～２１４はマルチプレクサ、３０１～３１０はインバータ、４０１～４０３は０Ｒゲートであり、a_1～a_8は８桁の被開平数入力、q_1～q_4は４桁の平方根出力、r_1～r_5は５桁の残余出力である。

実行手順に従い、加算はフルアダーで行ない、その結果、最上位の桁上げの値により加算結果と前段の結果のどちらかを用いるかの選択はマルチプレクサを用いている。

このハード構成において$a_1a_2+11=k_1k_2k_3$の計算でk_1k_2はa_1+a_2+1に一致することを利用してフルアダー１０１の入力に"１"とa_1とa_2を入力して結果を得る。またk_3は$\overline{a_2}$であるのでインバータ３０１でa_2を反転している。k_1は出力q_1として用いられ、k_2k_3かa_1a_2かの選択はマルチプレクサ２０１及び２０２を用いて制御信号k_1により行なわれる。k_1が"１"のときk_2k_3が選ばれ、k_1が"０"のときa_1a_2が選ばれs_1s_2として次段で用いられる。次段では、$s_1s_2a_3a_4+1\overline{q_1}11=m_1m_2m_3m_4m_5$の計算が行なわれる。$a_3a_4+11$の計算は初段と同様に行なわれ、その桁上げ$c_1$と$s_2$と$\overline{q_1}$との加算がフルアダー１０２で行なわれる。この加算結果の桁上げc_2とs_1と"１"との３入力の加算が必要となるが、１つの入力が"１"であるのでこの加算結果の桁上げm_1はc_2とs_1との論理和で計算されるのでc_2とs_1を入力とする０Ｒゲート４０１で作られる。m_2はq_2が"１"のときのみ次段で選択され用いられるべきであるが、m_1（$=q_2$）が"１"のときは必ず"０"となることが以下で示すようにわかるので必要でない。

２ｎ桁の２進数の平方根はｎ桁で表され、その残余が最も大きくなるのは、２ｎ桁の２進数の最大値$2^{2n}-1$の平方根2^n-1が算出される場合であり、その残余は、$2^{2n}-1-(2^n-1)^2=2^{n+1}-2$となり、$2^{n+1}$を超えない。従って

m_1＝"１"のとき$m_2m_3m_4m_5$は残余となるが、上記ｎ＝２の場合に相当し、$2^5=1000$を超えない。即ち$m_2=0$となる。

次段ではm_1により$s_2a_3a_4$と$m_3m_4m_5$のどちらかが選択され計算の対象となり、以下同様に第４図に示す計算を行なうことにより、平方根q_1～q_4及び残余r_1～r_5を求めることができる。

また、第１図ではフルアダー１０個及びマルチプレクサ１４個，０Ｒゲート３個，インバータ１０個で開平装置を構成している。フルアダーを６個のゲート及びマルチプレクサを４個のゲートで構成すると総ゲート数は１２９個となる。ＣＡＳを使用した場合の１４３個に比べ１４個のゲート数が削減される。さらに必ず１を入力するフルアダー等でゲートの節約及び残余を必要としない場合、最下段のマルチプレクサの節約が可能である。演算速度はフルアダーの通過回数を考慮すると本演算方式では最長ルートで７回であり、ＣＡＳを使用する方法がシリアル的に２０回であり、２倍以上の速度改善がなされる。２ｎ桁の２進数からｎ桁の２進数として開平演算をする場合、ゲート数は、ＣＡＳを使用したものが$7n^2+8n-1$個に対し本発明では$\frac{11}{2}n^2+\frac{21}{2}n-1$個、フルアダー通過回数は、ＣＡＳを使用したものが$n^2+n$回であるのに対し、本発明では$\frac{1}{2}n^2-\frac{1}{2}n+1$回となり、ｎが大きくなると素子の削減割合は$\frac{5}{14}=21.4$％、フルアダー通過回数は$\frac{1}{2}$回に近くなる。

発明の効果

以上述べたように、本発明によれば、開平演算を並列演算方式で実現する場合、素子数の削減と演算速度は２倍以上にすることができ、この実用的効果は大きい。

4、図面の簡単な説明

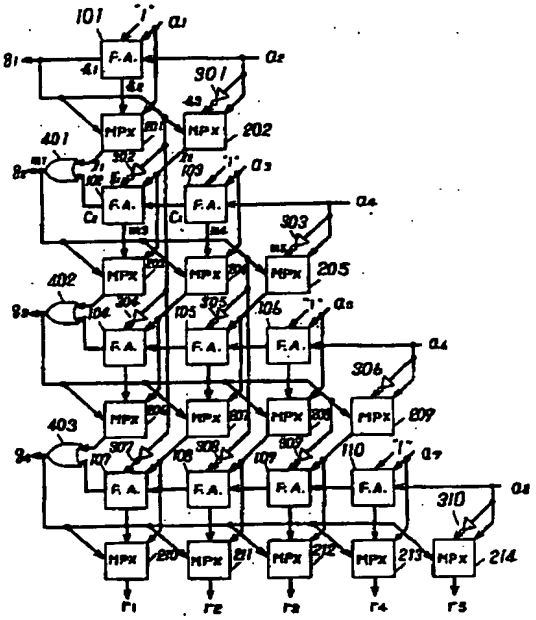
第１図は本発明の開平演算方式の一実施例のブロック図、第２図は本発明の非回復型の平方根の導出アルゴリズムのフローチャート図、第３図は従来の一般的な非回復型の平方根の導出アルゴリズムのフローチャート図、第４図は本発明における開平演算方式の実行手順の筆算形式による説明図である。

１０１～１１０……フルアダー、２０１～２１４……マルチプレクサ。

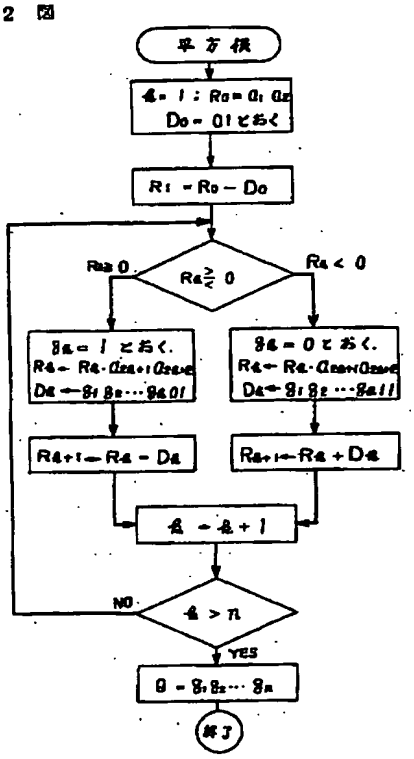
代理人の氏名　弁理士　中　尾　敏　男　ほか１名

101～110 … フルアダー（F.A.）
201～214 … マルチプレクサ（MPX）

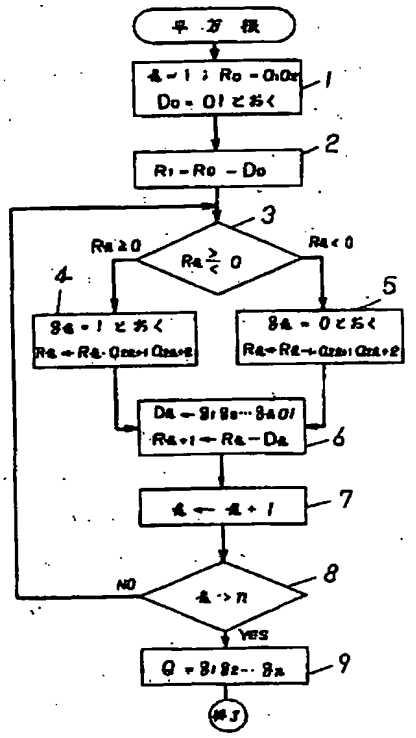
第１図



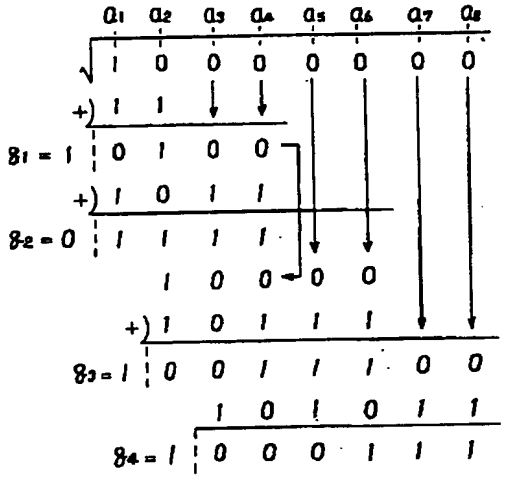
第２図



第３図



第 4 図


a1 a2 a3 a4 a5 a6 a7 a8
1 0 0 0 0 0 0 0
+) 1 1
g1 = 1 0 1 0 0
+) 1 0 1 1
g2 = 0 1 1 1 1
1 0 0 0 0
+) 1 0 1 1 1
g3 = 1 0 0 1 1 1 0 0
1 0 1 0 1 1
g4 = 1 0 0 0 1 1 1